UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Cheng-Han Chung

Audio Transformations Based Explanations (ATBE) for deep learning models trained on musical data

Master's Thesis (30 ECTS)

Supervisor: Anna Aljanaki

Tartu 2024

Audio Transformations Based Explanations (ATBE) for deep learning models trained on musical data

Abstract:

Explaining deep learning model behaviour is difficult. For computer vision models, a number of methods exist, which can highlight the areas that the network focuses on in an image. For music classification model, this does not usually result in satisfactory result as interpreting models trained on audio has to be based not on visual but on musical concepts, related to acoustic characteristics important to humans, such as pitch, tempo, melody, harmony. In this thesis we propose a concept of audio transformations based explanations (ATBE) for deep learning models for music. We release a python package called ATBE, which can explain which acoustic properties are important for predicting certain classes using error analysis on modified input, and using LIME on surrogate features created in the process of augmentation.

Keywords: music information retrieval, model explainability, data augmentation

CERCS: P176-Artificial Intelligence, P170-Computer science, numerical analysis, systems, control

Heli muutuse abil süvõppe mudelite selgitamine

Lühikokkuvõte: Süvaõppe mudeli käitumise selgitamine on keeruline. Arvutinägemismudelite puhul on olemas mitu meetodit, millega saab esile tuua piirkonnad, millele võrk pildil keskendub. Muusika klassifitseerimise mudeli puhul ei anna see tavaliselt rahuldavat tulemust, sest heli põhjal treenitud mudelite tõlgendamine peab põhinema mitte visuaalsetel, vaid muusikalistel mõistetel, mis on seotud inimeste jaoks oluliste akustiliste omadustega, nagu helikõrgus, tempo, meloodia, harmoonia. Käesolevas lõputöös pakume välja uut meetodit, mis aitab heli muutes välja selgitada, millised akustilised omadused olid olulised teatud klasside ennustamiseks. Selleks kasutatakse neid vigu, mida mudel muudetud sisendil teeb, ja LIME meetodi.

Võtmesõnad: muusika andmeanalüütika, mudeli seletatavus

CERCS: P176-Tehisintellekt, P170-Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

List of Abbreviations

ADAM	Adaptive Moment Estimation
AITAH	Audio Integrated Transformation Hub
ATBE	Audio transformation based explanations
BPM	Beats Per Minute
CNN	Convolution Neural Network
DRC	Dynamic Range Compression
FFT	Fast Fourier Transform
HPSS	Harmonic/Percussive Source Separation
LIME	Local Interpretable Model-Agnostic Explanations
MIR	Music Information Retrieval
OS	Operation System
PALUN	Presenting Audio by Illustrating Units
SHAP	SHapley Additive exPlanations
SOTA	State-of-the-Art
STFT	Short-time Fourier transform
WSL	Subsystem for Linux

List of Tables

List of Figures

Contents

Acknowledgement

I appreciate the support and guidance I obtained while studying at the University of Tartu. My deepest gratitude goes to my supervisor, Professor Anna Aljanaki, for her invaluable feedback and unwavering assistance in my thesis work. Her knowledge and perspective significantly enhanced my learning and research.

I extend my thanks to all who stood by me during my academic journey, for their support, belief in my abilities, and assistance in reaching this important milestone.

1 Introduction

1.1 Research Background

Music is the spice of life, it adds flavor in our daily life. One song can make your day, for instance, arousing your memories about a certain moment (e.g., your proposal). Music also is a way of perceiving the world and an instrument of knowledge. Scholars study music by exploring its impact on society from various perspectives. Music Information Retrieval (MIR) plays a crucial role in making music collections accessible to listeners. MIR is an interdisciplinary field that combines art and technology, focusing on extracting meaningful features from music, indexing music, and developing search and retrieval algorithms. MIR utilizes digital music data for tasks like information retrieval, classification, and sequence labeling, incorporating machine learning and human expertise. MIR is essential for enhancing music accessibility, analysis, and user interaction in today's society.

Deep-learning models have achieved significant breakthroughs in various fields including computer vision, speech recognition, and natural language processing, they have become the benchmark for MIR tasks such as music classification, recommendation systems, source separation, instrument recognition, transcription, and generation. However, due to their complex structure and a huge number of parameters, which can be in the millions, deep-learning models are often seen as "black box" systems. This complexity makes it difficult to understand their decision-making process, potentially leading to unexpected learning outcomes and decisions based on irrelevant or misleading information. This lack of transparency raises questions about what these systems are truly learning.

As deep-learning models become involved in the musical applications or services, the need for transparency and interpretability grows [JHD23]. It has been shown, that music information retrieval systems are prone to taking shortcuts by learning irrelevant properties in the signal instead of the relevant ones that are more difficult to learn [Stu14]. The interpretability allows stakeholders, including musicians, sound engineers, machine learning practitioners, and corporations, to trust these systems to produce reliable results and understand how decisions are made. For example, if your music application always recommends you music that you are not interested in, may be it is biased, as it has been shown for some machine learning based music recommendation systems [CTMG23]. Interpretable explanations of these models would be extremely helpful for identifying such biases, ensuring fair treatment of all users, and maintaining trust.

While writing this thesis, the author used ChatGPT to correct language inaccuracies. The generative model contributed to improving the text and correcting grammatical and syntactic errors.

1.2 Problem statement and Solution

In music-related applications, providing faithful, human-understandable explanations of model predictions can increase trustworthiness and enhance user experience. From a developer's perspective, an interpretable model could better reveal potential issues, allowing the detection of biases, malfunctions, or possible adversarial attacks. Interpretability can also provide insights into the target problem, contributing to the advancement of the field as a whole.

Hence, the demand for interpretability has led to significant advancements in explainable AI, which has become an important component of all AI sub-fields. Interpretable deep-learning model aims to clarify complex models, building trust in these models. The goal is to make the decisions of deep-learning model understandable, providing insights into their learning processes, ensuring that they align with their intended purpose. However, despite the preference to design models with interpretability, there will always be models without any internal interpretability mechanisms, necessitating external explainability tools.

Several tools and methods from the field of interpretable machine learning have found their way into MIR. However, those solutions lack musically meaningful and intuitive concepts. We introduce a method to interpret the influence of acoustics features by error analysis and using Local Interpretable Model-Agnostic Explanations (LIME). LIME have been proposed as a technique to enhance the interpretability of black box Machine Learning (ML) algorithms. LIME aims to provide explanations for individual predictions by creating simpler interpretable models around them, typically using random perturbation and feature selection methods. However, the randomness in perturbation can lead to instability in explanations, hindering deployment in sensitive domains [ZK21]. To address this issue, we propose a controllable feature perturbation method.

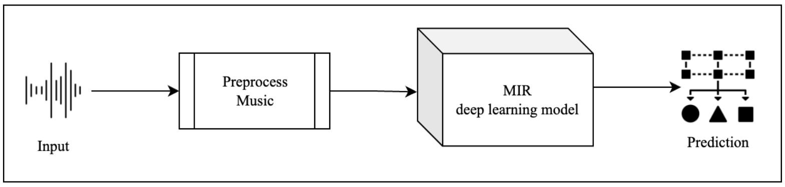


Figure 1. MIR models

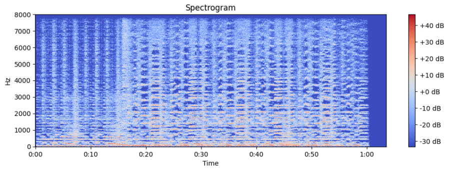


Figure 2. Example of a spectrogram

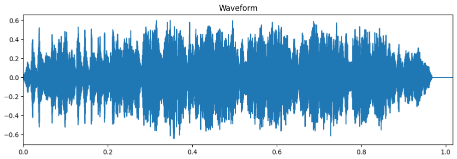


Figure 3. Example of a sound wave (time series of amplitude values).

2 Background and Related Work

In this section, we explore the difficulties related to interpretability that arise with State-of-the-Art (SOTA) models in MIR tasks, along with the prevalent techniques currently used to explain deep-learning models.

2.1 Deep-learning models in MIR task

MIR encompasses various tasks aimed at extracting meaningful information from music data. The fundamental MIR tasks include:

- music classification, for example categorizing music by genre [Bah18] or mood [DHP+18]

- source separation, which is extraction of individual sound sources (typically corresponding to individual instruments or instrument groups) from a mixed (mastered) audio recording [HKVM20]. Melody extraction is a related task [KN19].

- instrument recognition, which is classifying audio files by the instrument that plays it [SP22]

- music recommendation, which is suggesting music to users based on their preferences [SFR21]

- chord estimation [WCNY20], which is the process of recognizing musical chords that are attributable to segments of a music piece.

- auto-tagging [WFBS20], which is automated process of assigning descriptive tags or labels to music tracks without human intervention.

- beat detection [dSMB21], which is the process of identifying and tracking the rhythmic beats in music signals

- music transcription, which is converting music into symbolic notation, such as score [NNGY21].

From a physical point of view, music consists of sound waves - vibrations of air, which are repetitive and organized to form pleasing compositions [SS18]. Sound is characterized as waves that carry energy through a medium without moving the medium itself. Such waves are called longitudinal, where particles compress and decompress in the same direction as the wave's propagation.

Deep learning models learn to extract representations from a signal, that represent all the important music properties. They can be trained using either time series of raw amplitude samples, or a time-frequency representation (usually, in case of music, a spectrogram) (see Fig. 1).

Spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time (Fig. 2). Spectrogram shows the intensity of different frequencies in the signal over time. A deep-learning model can extract musically meaningful features from such data, such as timbre, instruments, onsets, pitch height.

Raw samples of amplitude refer to the direct measurements of the strength or intensity of the audio signal at different points in time (Fig. 3). In the context of music, raw samples of amplitude represent a signal of mixed frequency before frequency analysis, or a sound wave. These samples are typically captured at regular intervals, known as the sampling rate, and are represented as a sequence of numerical values. Because a sampling rate is usually 44500 samples per second, these time series are unusually long, much longer than sequences used in natural language processing, for example.

Spectrograms provide a visual representation of the frequency content of audio signals over time, offering a more comprehensive and detailed source of information for analysis which is easier to train on than raw audio signal. This richer data in spectrograms allows for more accurate and nuanced processing and interpretation of audio data compared to working with waveforms alone, which would require more training data. Won, Minz, et al. [WFBS20] conducted a comparison and benchmarked the SOTA model used on spectrogram-based and raw samples of amplitude-based data. The findings indicated that the spectrogram-based model outperformed the raw samples of amplitude-based model when a dataset is not big enough.

In this paper, a Convolution Neural Network (CNN) spectrogram-based music genre classification model will be developed in Section 4 to showcase the practical application of deep learning in the field of ATBE.

2.2 Interpretability in machine learning

Interpretability refers to how well a human can comprehend the reason behind a decision. The reason interpretability is necessary stems from a lack of completeness in problem formalization, indicating that merely obtaining a prediction is insufficient for certain problems or tasks. It is crucial for the model to elucidate the reasoning behind the prediction, as a correct prediction only partially addresses the original issue. For instance, if your music streaming service consistently suggests songs that do not align with your preferences (such as recommending a heavy metal track like "Welcome to Hell" by Venom when you dislike heavy metal), it could potentially drive away customers. Similarly, within the academic realm, while the aim of science is to acquire knowledge, many issues are resolved using extensive datasets and opaque machine learning models. In such cases, the model itself serves as the knowledge source rather than the data. Interpretability enables the extraction of the additional knowledge encapsulated by the model.

Biases often are acquired by deep learning models from the training data, potentially leading model to discriminate against underrepresented groups. The capability of interpretation serves as a valuable debugging tool for detecting bias in deep learning models. Even in low-risk scenarios like music recommendations, interpretability holds value during research, development, and post-deployment phases. In case an underperforming model is used in production, an interpretation of an incorrect prediction aids in understanding the root cause of the mistake. If a machine learning model can provide explanations for decisions, it facilitates the assessment of certain characteristics such as fairness, privacy, reliability, robustness, causality, and trust.

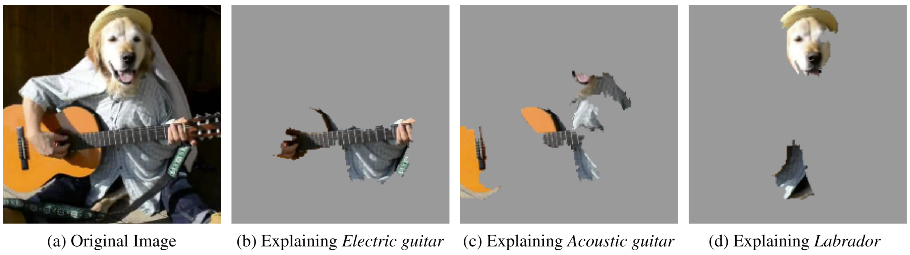
2.2.1 Categorization of Interpretability Methods

Methods of interpretability can be categorized according to various criteria, which can be:

- **Intrinsic or post hoc.** Intrinsic interpretability pertains to machine learning models seen as interpretable because of their straightforward design, like decision trees or linear models. Post hoc interpretability involves employing interpretation methods after model training.

- **model-specific or model-agnostic.** Model-specific interpretation tools are restricted to certain model classes, such as explanation of regression weights in a linear model, as the explanation of models that are inherently interpretable is always model-specific. Methods that are designed solely for interpreting neural networks are also specific to those models. Model-agnostic tools can be applied to any machine learning model and are used after the model has been trained (post hoc). These agnostic methods typically analyze pairs of feature input and output. Due to their nature, these methods do not have access to internal model details like weights or structural information.

- **Local or global.** The local interpretation method explains a single prediction, while the global interpretation method elucidates the overall model behavior.



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4. Example of explaining image classification model using LIME. [RSG16]

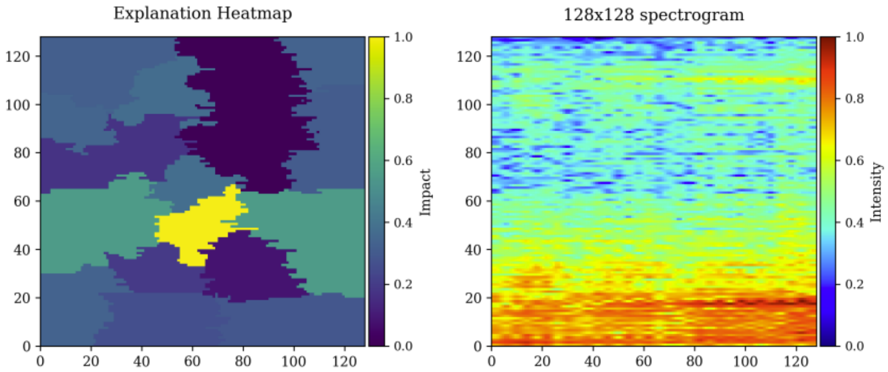


Figure 5. Example of explaining a classification model trained on spectrograms. [SSB23]

2.2.2 Interpretability in MIR model

In the field of MIR, a deep learning model input can consist of images (spectrograms), raw audio sequences of amplitude values, or symbolic music (typically, encoded as text). All of these inputs are challenging to interpret.

Sturm et al. [Stu14] proposed to investigate whether an MIR system's performance is truly based on musical knowledge or if it is exploiting dataset characteristics. The method involves demonstrating how MIR systems can achieve high figures of merit without utilizing musical knowledge, indicating the need for improvement in MIR systems and evaluation methods.

It is challenging to apply image interpretation methods effectively in MIR task models. Spectrogram-based models pose difficulties in interpreting musical meaning and intuitive concepts. This complexity underscores the need for further research and development in the field to enhance the interpretability and effectiveness of MIR models.

Researchers are trying to provide solutions to address interpretability challenge in MIR model. Choi, Keunwoo, et al. proposed a method called Auralisation for interpreting deep convolutional neural networks (CNNs) applied to spectrograms [CFSK15].

It can explain a single example prediction, using deconvolution across the layers of neural network. A listenable explanation is provided in the end, which is very difficult to interpret by hearing because it is based on neural network learned patterns which might not necessarily be related to musical concepts but rather to some parts of the spectrum it focused on. Alonso-Jiménez, Pablo, et al. introduced PECMAE, an interpretable model for music audio classification based on prototype learning, leveraging pre-trained autoencoders for enhanced interpretability [AJPBR$^+$24]. It utilizes a diffusion decoder for sonification of prototypes, eliminating the need for direct dependency on specific training samples for prototype reconstruction. Foscarin, Francesco, et al. [FHP$^+$22] suggested a supervised approach allowing users to specify a musical idea to examine its impact on the model's choices, facilitating the comprehension of the model's music-related thought process. Additionally, it presents an unsupervised technique that automatically detects and displays significant musical ideas identified by the model, aiding users in uncovering the model's key musical aspects. This technique works for symbolic music only. Haunschmid et al. [HMW20] suggest audioLIME, which improves interpretability through the utilization of source separation to generate understandable and audible elements from audio data, going beyond conventional spectrogram-based explanations. Because this technique is based on LIME, it can only explain a predictions one by one. Rao et al. [RD$^+$22] introduced a method that breaking down how the model makes decisions or predictions by visualizing what the model focuses on when extracting melodies from polyphonic music.

A method proposed in this thesis is a mixture of approaches used in [Stu14] and [HMW20]. To bridge the gap of interpretability in MIR task deep learning model, we introduced a local post-hoc, model-agnostic example-based explanation method. This method can interpret importance of musical features by using error analysis on modified inputs to a model. In local model-Agnostic method, a prominent example of local methods is Local interpretable model-agnostic explanations (LIME). In traditional LIME [RSG16], interpretability is achieved by perturbing input features X randomly and observing model behaviour. LIME can be adapted to computer vision by replacing features with individual pixels and observing their importance to prediction. This method provides intuitive instance-based explanations for image classification models. For example, Rebeiro et al. [RSG16] implemented this method to explain image recognition neural network. Figure 4 shows how the parts of an image that a model was focusing on when making a decision are highlighted. For tasks such as classifying non-intuitive images like spectrograms [SSB23], it's not easy for a human to understand how the model is working using the LIME explanation, because the highlighted parts of the image might not be intuitively understandable. Hence, in our propose, we try to create musically meaningful features that could be used as input to LIME.

3 ATBE method

In machine listening tasks, state of the art models are not trained on low-level features extracted using signal processing anymore, but on raw data input (either in a form of spectrograms or amplitude samples sequence).

Even in the early days of music information retrieval when the spectral features were used, those were oftentimes low-level features that were not easily connected to human-understandable concepts either [AB12]. Extracting human-interpretable features, let alone modifying them, is a huge challenge. For instance, identifying melody, or chords is a challenge in itself [RD$^+$22].

To add interpretability, we propose modifying audio input directly by manipulating the audio file using transformations. These transformations can be both irrelevant augmentation type transformations, such as dynamic range compression, small pitch shift and time stretch, and relevant data distribution modifying transformations such as mode, removing stems from audio, changing tempo significantly.

In this thesis, we show how this method works on audio data, but it can also be applied to symbolic data, and transformed symbolic data can be synthesized into audio.

For example, we can observe how a model will react when all fiddle parts (characteristic of folk music) would be replaced by saxophone parts (characteristic of jazz music).

3.1 Method description

Transformations created using ATBE method could be used both for creating LIME local explanations and for global model-agnostic explanation. We can observe the errors that a model makes, when its input audio is augmented in a musically meaningful way, and make conclusions about what features a model bases its decisions on when predicting a certain class.

The properties that could be changed in this way are numerous: modifying a key from major to minor, replacing instruments with different ones (for instance, using style transfer), changing pitch, tempo, loudness. The possibilities are limitless, in this thesis a small subset of these possible techniques is implemented.

Let’s make an example to understand the intuition behind such modifications. If a model is classifying instruments, its performance should not degrade when some minor tempo modification is made in the input. However, it should react to pitch shift, as instrument tessitura is usually fixed up to a certain degree. However, if a model is doing melody detection, it should not be confused when a whole input is pitch shifted.

In this thesis we will demonstrate the method by modifying some of the easier properties to modify: pitch, tempo, dynamic range compression rate, and removing the percussive component by using harmonic-percussive source separation. By demonstrating how the method works this way, we open a path to add more transformations that

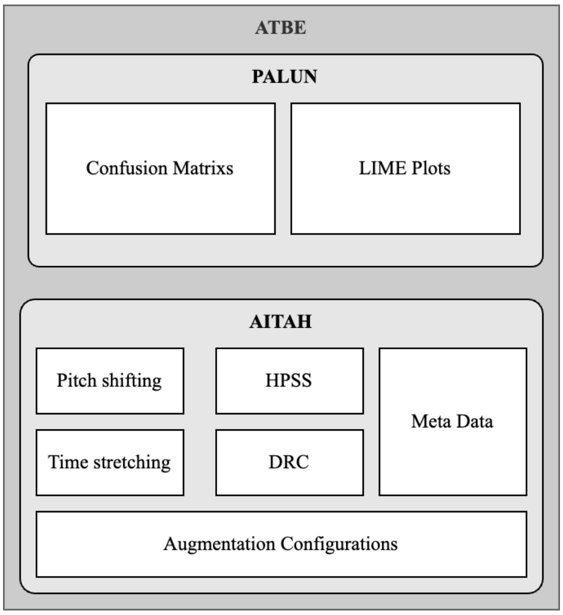


Figure 6. ATBE Framework

could be model-specific.

3.2 Implementation

ATBE is a Python library that has been designed to aid users in explaining their music classification models, with a particular focus on detecting undesirable effects learned by the model, that oftentimes happens when a model is trying to cut corners by learning easier concepts instead of the more musically meaningful ones (for example, dynamic range compression could serve as a proxy for genre detection).

The ATBE library is divided into two primary components (See Fig. 6). Firstly, we have Audio Integrated Transformation Hub (AITAH), a tool that augments a provided audio dataset. It does this by applying a range of different transformations to the original audio files.

The second component, Presenting Audio by Illustrating Units (PALUN), serves as a visualisation tool that displays the model's prediction results. It plays a crucial role in demonstrating the influence of various features on the model. By highlighting these influences, it helps users to understand better how their model is making its decisions, providing valuable insights into the feature importance.

3.2.1 AITAH (Audio Integrated Transformation Hub)

This component is partially based on MUDA library [MHB15a] and provides various customizable audio transformations through the establishment of customizable settings using a YAML file (See Fig. 7) which supports the following transformations:

1. Pitch shifting (modifies the frequencies in the audio).
2. Time stretching (allows to alter the audio duration without affecting its pitch, therefore affecting only tempo).
3. Dynamic range compression (reduces the volume of loud sounds and amplifies quiet ones, therefore reducing the dynamic range). This is a common audio processing method in music production.
4. Harmonic/Percussive Source Separation (separates harmonic and percussive elements of the audio). This allows to remove the percussive component to create a binary feature (with/without drums).

The module can save metadata for managing and retrieving transformation information about the audio file.

In deformations of Pitch Shifting, Time Stretching, and Dynamic Range Compression (DRC), we use MUDA [MHB15b], a python library for musical data augmentation, which is implemented to expand training sets by incorporating musically relevant modifications to audio while preserving annotations. For Harmonic/Percussive Source Separation (HPSS), we use librosa [MRL$^+$15], a Python package designed for audio and music signal processing, offering tools for analyzing music and audio signals with ease.

- **Augmentation Configurations**

The personalized configuration process for the AITAH component is achieved through the usage of a YAML file. The configuration YAML file is a powerful and flexible tool that allows users to precisely tailor the audio transformation pipeline to their specific requirements and preferences.

Key parameters that can be adjusted include the upper and lower boundaries for pitch shifting, the selection from a number of different time stretching techniques, the choice of dynamic range compression styles, or the application of Harmonic/Percussive Source Separation.

This provides an efficient way to manage the information about the audio augmentations applied to each individual audio file, as well as retrieve this information when required.

For examples of how these configurations might look, see appendix 6. The example provides a clearer illustration of how the YAML file can be edited and structured to achieve the desired results.

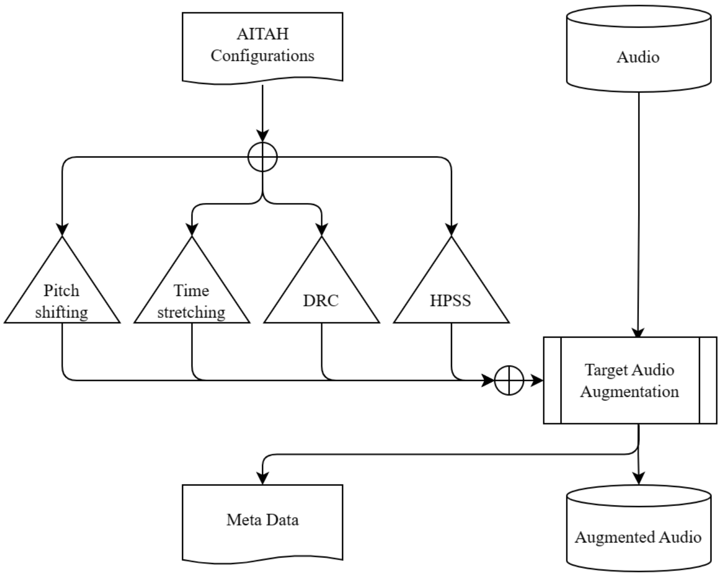


Figure 7. AITAH Pipeline

- **Pitch shifting**

Pitch shifting is a technique used in music production and audio engineering to modify the pitch of a sound while keeping its duration constant [Yük22]. Different common techniques for pitch shifting include linear Pitch shifting, non-linear Pitch shifting, and granular Pitch shifting.

Linear pitch shifting is a precise method that allows for smooth and gradual adjustments in pitch while maintaining a constant tempo [KMT08]. It ensures seamless transitions between pitches, resulting in natural and coherent modifications to the audio content. This method is commonly used in audio applications where maintaining the original tempo is crucial, such as music production and vocal correction.

Non-linear pitch shifting involves altering the pitch of audio signals in a non-linear manner, resulting in more abrupt and noticeable pitch changes [TYS+22]. While this method can create unique and dramatic effects, it may not always provide smooth transitions between pitches, leading to potential artifacts or unnatural sounds.

Granular pitch shifting breaks down the audio signal into small "grains" and manipulates them to achieve pitch changes [EMO+20]. This method allows for more complex and experimental pitch modifications but may introduce artifacts or distortion in the audio.

In ATBE, we choose linear pitch shifting as the preferred method due to its precision, smooth transitions between pitches, and ability to maintain a constant tempo. Linear pitch shifting ensures that the pitch modifications are natural and seamless, improving the overall quality and coherence of the audio content. By utilizing linear pitch shifting, we aim to achieve flawless pitch adjustments while preserving the integrity of the original audio, making it a suitable choice for our audio processing needs.

- **Time stretching**

Time stretching is a technique used in music production and audio engineering to alter the duration of an audio signal without affecting its pitch [FWVH23]. Time stretching allows users to change the speed of the audio playback, making it faster or slower, while keeping the original pitch intact. Different common techniques for time stretching include Phase Vocoder, Time Domain Processing, Granular Synthesis, and Logspace Time Stretching.

The Phase Vocoder technique involves dividing the audio signal into short overlapping segments called frames [PH17]. Each frame is analyzed using the Fast Fourier Transform (FFT) to extract the frequency content. By stretching or compressing the time axis of these frames, the audio signal's duration can be adjusted without changing its pitch. This technique maintains audio quality by preserving phase relationships between different frequency components.

Time Domain Processing involves directly manipulating the audio signal in the time domain [FWVH23]. An example of this is the Time Domain Harmonic Scaling (TDHS) technique, which modifies the time gaps between harmonic elements to either elongate or shorten the audio signal. By using this approach, the original harmonic pattern of the audio is maintained even as its length is adjusted.

Granular Synthesis breaks down the audio signal into small grains or segments [BEH20]. By manipulating the playback speed and overlapping these grains, the duration of the audio signal can be altered. This method allows for creative time stretching effects and can produce unique textures and timbres in the audio.

Logspace Time Stretching is a method that alters the duration of an audio signal by stretching or compressing it in a logarithmic time scale [DV17]. This technique allows for more natural-sounding time stretching by focusing on adjusting the time intervals in a logarithmic manner, which can result in smoother transitions and less audible artifacts compared to linear time stretching methods.

In ATBE, Logspace time stretching is selected for its capacity to adjust the length of an audio signal by stretching or compressing it on a logarithmic time scale. This technique enhances time stretching realism by focusing on logarithmic changes in time intervals, resulting in smoother transitions and fewer noticeable artifacts compared to linear methods. The use of a logarithmic time scale in logspace time stretching is favored for its ability to deliver high-quality time stretching effects while preserving the integrity and coherence of the audio content.

- **Dynamic Range Compression (DRC)**

DRC is a technique used in audio processing to reduce the dynamic range of an audio signal [MKD18]. DRC is commonly utilized to modify the loudness of an audio file by reducing its dynamic range. The dynamic range refers to the difference between the loudest and quietest parts of an audio signal. By applying time-varying gain signals to the input audio based on desired output ranges, DRC works by boosting the softer parts of the audio signal and attenuating the louder parts. This process results in a more balanced and consistent audio output where the volume levels are more uniform throughout the audio file. DRC helps in making the audio more consistent in terms of volume levels, ensuring that all parts of the audio signal are audible. DRC is usually not a musically meaningful feature, so a model exclusively focusing on this feature is not likely to have learned anything useful. This features serves as a negative marker of quality.

- **Harmonic/Percussive Source Separation (HPSS)**

HPSS is a technique used in audio signal processing to separate the harmonic (tonal) components from the percussive (transient) components of an audio signal [RPM+23]. The goal of HPSS is to isolate the musical elements in a sound recording that correspond to pitched instruments (harmonic) from those of non-pitched instruments or sounds (percussive). By separating these components, HPSS enables a more detailed analysis of the underlying musical structure of the audio signal. When interpreting a music classification model, implementing HPSS (Harmonic-Percussive Separation) could help understand which components the model focused on when predicting, similarly to audioLIME method [HMW20] but both in a local and global model-agnostic way. By removing the percussive elements and only keeping the harmonic components of the audio input, HPSS allows for a focused analysis of the melodic and harmonic content of the music [ŞJEC11].

- **Meta Data**

In the process of augmenting audio files, each file is accompanied by its own meta data, which includes specifics such as the type of augmentation applied, the number of semitones by which the audio is shifted, or the extent to which the time stretch rate is altered. Subsequently, these meta files are aggregated and utilized by the PALUN component to facilitate the interpretation of model predictions effectively.

3.2.2 PALUN(Presenting Audio by Illustrating Units)

This component is designed to visually display information on high-level features impact the model's prediction. The augmented audio data from the AITAH component are fed into a model that we are trying to explain, to generate prediction results. Lastly, the visualization function presents information in the form of a confusion matrix and a LIME plot. For a scheme of the PALUN pipeline, see Fig. 8.

The confusion matrix provides a comprehensive breakdown and comparison of the predicted and actual classification outcomes, thereby allowing for an in-depth evaluation of the model's performance. The LIME plot offers a detailed analysis of the features'

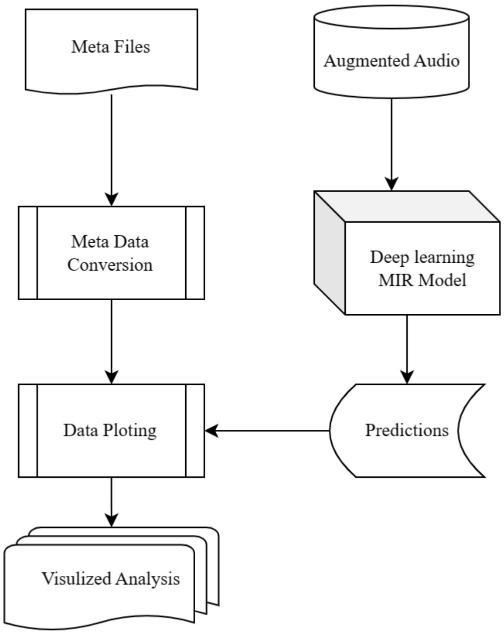


Figure 8. PALUN Pipeline

influence on individual instances. By examining those plots, we can gain a better understanding of which features had most impact in the context of the model's decision-making process.

It could be theoretically possible to use global model-agnostic explanations with this method, if feature that was modified can be converted from relative values to absolute ones. It is possible for some types of data, such as mode (minor/major), because it is an objective property. It is however difficult for other transformations, such as time stretch or pitch shift.

For example, time stretch modifies both tempo and onset density of the file. We could measure Beats Per Minute (BPM) to provides an objective, quantifiable measure for consistent comparison across samples, but we are still not measuring the same thing that we modified, we measure it in an indirect imprecise way (there is an error associated with beat detection methods).

Confusion Matrix

We create a confusion matrix to offer a detailed summary and juxtaposition of the actual and predicted classification results. This illustrative tool enables us to perform a thorough assessment of the model's accuracy, providing insights into the precision of the predictions and the nature of the errors committed.

LIME Plots

LIME plots offer visual representations that illuminate the features that are most influential in a deep learning model's predictions for individual instances. By examining these plots, users can acquire a deeper comprehension of the model's decision-making process, specifically how certain features contribute to it.

4 Case study: applying ATBE to a deep learning model of ballroom dance music classification

In this section we will describe an experiment that showcases how our proposed method works. In section 4.1 we will describe a chosen dataset and explain what are desired and undesired behaviours when training a model on such dataset. In section 4.2 we will describe implementing and training a CNN model on this dataset.

4.1 Data

To demonstrate ATBE, we create and train a multi-class music classification CNN model on a public dataset called Ballroom dataset [GKD$^+$06] containing ballroom dancing music extracted from this website. The dataset consists of 698 tracks, divided into 8 different dance types: Cha Cha, Jive, Quickstep, Rumba, Samba, Tanga, Viennese Waltz, and Waltz. Distribution over classes is shown in Table 1. Each audio duration is about 30 seconds, total duration of dataset is 20,940 seconds. Sample rate is 44.1 kHz, audio is in WAV format (*.wav).

The model trained on this dataset can learn a variety of musical properties: tempo, timbre, instruments, lyrics in various languages (audible phonemes). The tempo and

Table 1. Number of tracks in Ballroom dataset

Type	**# Tracks**
Cha Cha	111
Jive	60
Quickstep	82
Rumba	98
Samba	86
Tango	86
Viennese Waltz	65
Waltz	110
Total	**698**

rhythm are the most important properties that separate these classes in a musical sense. The rest of it - timbre, lyrics, instruments - might of might not be important at all.

4.1.1 Data cleaning and preparation

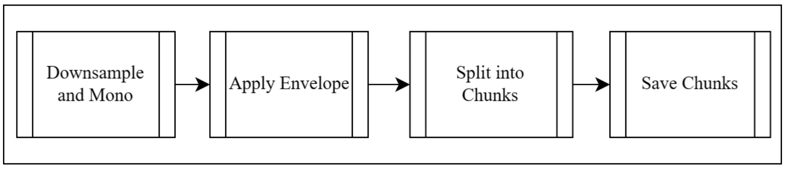


Figure 9. Data cleaning and preparation flow to train a CNN model

Audio files undergo a series of cleaning and preparation before moving on to machine learning tasks. Initially, the audio file is downsampled to 1.6 kHz, followed by conversion to a mono signal if necessary. Subsequently, a mask is created using a rolling maximum threshold to detect important audio segments and eliminate background noise with low energy. The threshold is established at 10. Finally, the audio is divided into consistent 5-second segments. The flow of data cleaning and preparation is described in Fig. 9.

4.2 Example CNN model

CNN is one of the most popular architectures in many music genre classification tasks, which often adopt a traditional to image processing architecture having cascading blocks of 2-dimensional filters and max-pooling, derived from well-known works in image recognition. We built and trained a CNN model to demonstrate the implementation scenario of ATBE, as described in Table 2 and Fig. 10.

4.2.1 Audio preprocessing

To train CNN model, mel spectrogram was used as input. It is a spectrogram where the frequencies are logarithmically spaced using mel-scale. The short-time Fourier transform (STFT) is computed by segmenting the waveform into overlapping frames of 255 samples, with a step size of 128 samples between consecutive frames, and applying a Fourier Transform of length 512 to each frame. This results in a two-dimensional array where one dimension represents time and the other represents frequency.

4.2.2 Setup

First, we split the Ballroom dataset into 75% training, 15% test, and 15% validation data. Then, we send audio file to the preprocessing. In the preprocessing stage, all audio is converted to mel-spectrograms in librosa and sent to the model.

We use DELL 13th Gen Intel i7-13700H, 14 cores, 20 logical processors for calculation, equipped with 32GB of memory. The Operation System(OS) is Windows Subsystem for Linux (WSL) with Ubuntu LTS 20.04. The Batch Size was set to 32, the total Epochs was 100, and the training time was about 10 minutes.

Adaptive Moment Estimation (ADAM) was utilized as the optimizer. It is a good choice for an optimizer with a dynamic learning rate.

4.2.3 Model training

The CNN model is built and trained for audio classification tasks using TensorFlow and Keras. Its architecture is tailored for processing normalized input spectrograms to aid in effective learning. We use 5 convolutional layers, each is followed by batch normalization for stability, max pooling for dimensionality reduction, and dropout to combat overfitting, and one fully connected layer. The model incorporates a flattening step to convert 2D feature maps into a 1D vector, with dense layers further processing these features into final class probabilities via softmax activation. Compiled with the ADAM optimizer known for its efficiency in sparse gradient handling, the model utilizes a learning rate of 0.001 and a weight decay of 1e-4 for regularization. `Sparse Categorical Crossentropy` is the chosen loss function suitable for multi-class classification with integer class labels. During training, the model uses a dataset split into training and validation sets, with a batch size of 32 and up to 100 epochs. To improve training efficiency and prevent overfitting, two callbacks are implemented: `Early Stopping` and `Reduce Learning Rate on Plateau`. `Early Stopping` monitors validation loss, stopping training if no improvement is observed over 10 epochs to conserve resources and avoid overfitting. The `Reduce Learning Rate on Plateau` callback adjusts the learning rate downward when validation accuracy stagnates for 5 consecutive epochs, aiding in fine-tuning near the loss landscape's minimum.

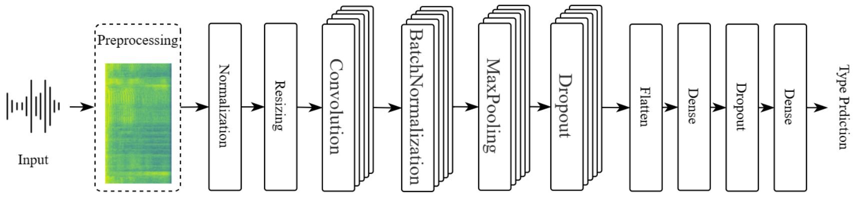


Figure 10. The architecture of example neural network model

Table 2. Configuration of the example CNN

Layer	Output Shape	Param #	Activation	Dropout
normalization	(None, 624, 257, 1)	3		
resizing	(None, 64, 64, 1)	0		
conv2d_1	(None, 64, 64, 32)	320	ReLU	
batch_normalization_1	(None, 64, 64, 32)	128		
max_pooling2d_1	(None, 32, 32, 32)	0		
dropout_1	(None, 32, 32, 32)	0		0.20
conv2d_2	(None, 32, 32, 64)	18,496	ReLU	
batch_normalization_2	(None, 32, 32, 64)	256		
max_pooling2d_2	(None, 16, 16, 64)	0		
dropout_2	(None, 16, 16, 64)	0		0.25
conv2d_3	(None, 16, 16, 128)	73,856	ReLU	
batch_normalization_3	(None, 16, 16, 128)	512		
max_pooling2d_3	(None, 8, 8, 128)	0		
dropout_3	(None, 8, 8, 128)	0		0.30
conv2d_4	(None, 8, 8, 256)	295,168	ReLU	
batch_normalization_4	(None, 8, 8, 256)	1,024		
max_pooling2d_4	(None, 4, 4, 256)	0		
dropout_4	(None, 4, 4, 256)	0		0.35
conv2d_5	(None, 4, 4, 512)	1,180,160	ReLU	
batch_normalization_5	(None, 4, 4, 512)	2,048		
max_pooling2d_5	(None, 2, 2, 512)	0		
dropout_5	(None, 2, 2, 512)	0		0.40
flatten	(None, 2048)	0		
dense	(None, 512)	1,049,088		
dropout_6	(None, 512)	0		0.50
dense_1 (Dense)	(None, 8)	4,104	softmax	

4.2.4 Model Result

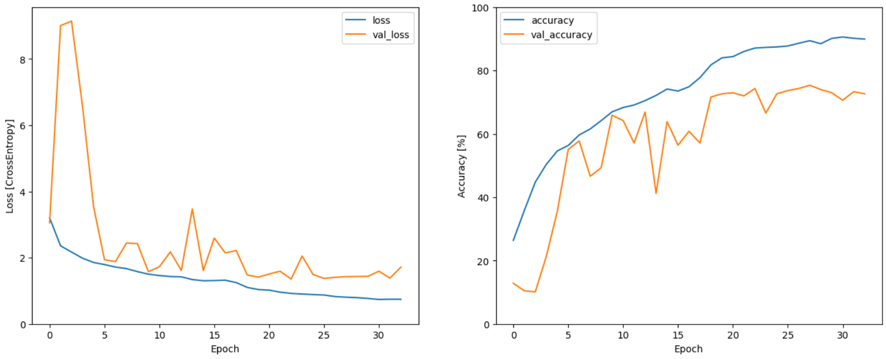


Figure 11. The accuracy and loss of the example CNN model

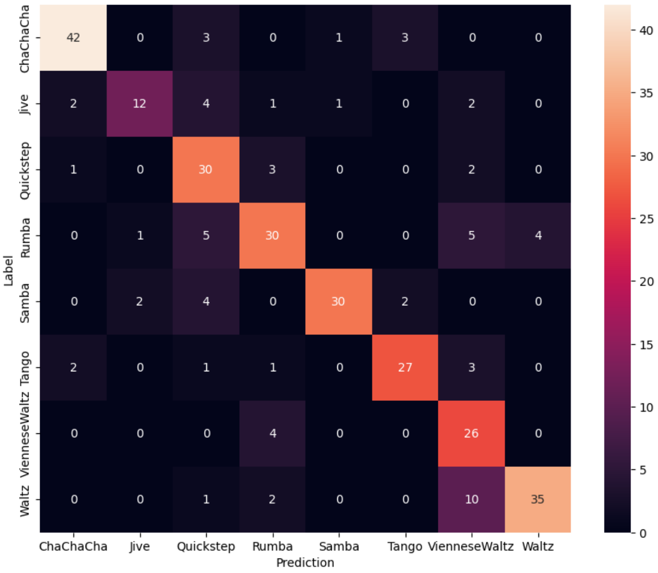


Figure 12. The confusion matrix of the example CNN model for the Ballroom test set.

The architecture proposed by this study has an accuracy of 89.86% in training dataset, 76.82% in test dataset, and a training loss of 0.76. The Fig. 11 shows that the model started to overfit after 15 epochs.

Fig. 12 illustrates a confusion matrix showing prediction of the CNN model on Ballroom test dataset containing 104 tracks. The model effectively differentiates between most dance styles. However, it struggles to accurately classify Waltz compared to other dance genres. While not flawless, this model is ready to showcase our ATBE method for model interpretation.

4.3 ATBE method on ballroom dance classification

Ballroom dances are heavily defined by tempo and rhythm, at least that's what a human expert would focus on when trying to predict them. There is however a specific sound to latin music or to classical waltzes, based on certain harmonies, traditions, and instruments performing the music. It is up to the researcher to decide whether, for instance, timbre modification should affect the model, or a model should recognize a latin dance for one even if it is performed by a gamelan. Because the interpretation is up to the model developer, we do not give any value assessment to the numbers but rather display them as they are. Model degradation after a certain transformation may be a good thing (for instance, we expect that a waltz will become Viennese waltz when increasing the tempo). It may be a bad thing (we do not expect our model to be affected by dynamic range compression).

In the training outcome of the sample model, there was confusion between waltz and Viennese waltz. Therefore, we will showcase the dataset containing waltz and Viennese waltz exclusively. This will provide a clear perspective on how ATBE assists in interpreting a music genre classification model.

4.3.1 AITAH pipeline implementation

The ATBE pipeline AITAH automatically generates augmented audio based on configuration. It proceeds to run a target classification model to predict the type of Ballroom music for all augmented audio generated by AITAH. Each augmented attribute for every audio will be stored in a YAML file, while the prediction result will be saved as a csv file.

Therefore, initially, we configure the AITAH pipeline in ATBE, as shown in Fig. 13. We apply Harmonic/Percussive Source Separation to eliminate Percussive Source. We adjust the time stretch to change the tempo, with the $upper$limit of tempo modification set at 1 and the $lower$ limit at -1. We define n_sample as 5 to generate 5 different tempo modifications within the specified bounds. Subsequently, we use pitch shift to alter the key, setting the $upper$ limit for key modification at 2 and the $lower$ limit at -2. Again, we set n_sample to 5 to create 5 distinct key modifications within the specified range. We configure DRC across six different real-world scenarios ($radio$, $filmstandard$, $filmlight$, $musicstandard$, $musiclight$, $speech$) to adjust audio compression settings. By applying the augmentation process in AITAH, we can acquire 2 (remove drums, or original audio) × 5(# of pitch shift) × 5(# of time stretch) × 6(# of DRC modification) augmented audio files for 2 different types of Ballroom dance music(waltz and Viennese waltz). Ultimately, we will have a total of 600 augmented Ballroom audio files.

```yaml
# Harmonic/Percussive Source Separation
hpss:
  apply: true

# Logspace Time Stretch
tempo_factor:
  lower: -1
  upper: 1
  n_samples: 5

# Linear Pitch Shift
keys:
  n_samples: 5
  lower: -2
  upper: 2

# Dynamic Range Compression
drc:
  - "radio"
  - "film standard"
  - "film light"
  - "music standard"
  - "music light"
  - "speech"
```

Figure 13. Configuration of AITAH in ATBE

Table 3. Accuracy Impact Summary

Metric	Viennese Waltz (%)	Waltz (%)
Original Accuracy	86.67	72.92
Accuracy on transformed audio	-33.94	7.81
Time Stretch	-2.67	-28.92
Pitch Shift	-6.67	-32.92
DRC	-6.67	27.08
HPSS	13.33	7.08

4.4 PALUN pipeline implementation

The PALUN pipeline of ATBE presents techniques for understanding the outcomes of how the model reacts to modified files, including accuracy impact summary table, confusion matrix, and LIME plot.

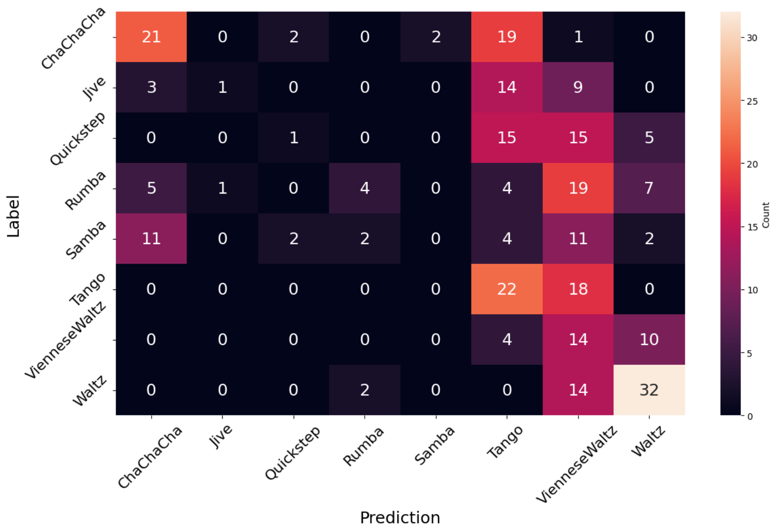
4.4.1 Ballroom dances accuracy and transformations

Through accuracy impact summary table provided by ATBE, it can offer an interpretive overview to elucidate the musical significance and intuitive understanding of model behavior across different musical attributes.

The accuracy impact summary table (Table 3) provides a comparative analysis of the accuracy metrics for Viennese Waltz and Waltz under various audio augmentation conditions. Initially, the original accuracy for Viennese Waltz is 86.67%, while for Waltz, it is 72.92%. When all types of augmentations are applied, Viennese Waltz experiences a significant accuracy drop of 33.94%, whereas Waltz shows a slight improvement with an increase of 7.81%.

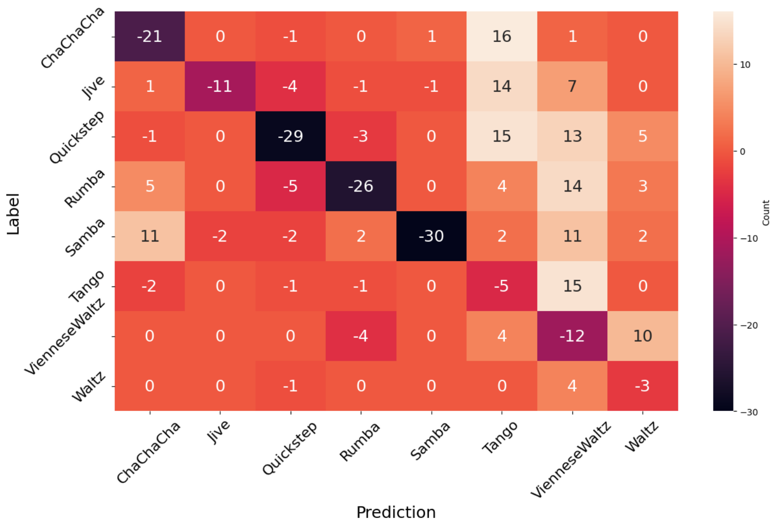
Time stretch (changing tempo) results in a minor accuracy decrease of 2.67% for Viennese Waltz and a substantial decrease of 28.92% for Waltz. Pitch shift (changing key) negatively impacts both dance styles, with Viennese Waltz decreasing by 6.67% and Waltz by 32.92%. Dynamic Range Compression decreases the accuracy of Viennese Waltz by 6.67%, but it significantly improves the accuracy of Waltz by 27.08%. Lastly, HPSS (removing the drums) increases the accuracy of Viennese Waltz by 13.33% and Waltz by 7.08%.

In summary, the table illustrates that different audio augmentations have varying impacts on the accuracy of classifying Viennese Waltz and Waltz. While some augmentations, like DRC, improve the accuracy for Waltz, others, such as Pitch Shift and Time Stretch, generally reduce the accuracy for both dance styles. Out of these transformations, only time stretch (modifying tempo) and HPSS should have possibly have an effect. Neither DRC nor slight pitch shift are in any way relevant to dance style classification in a musically meaningful way.



Label \ Prediction	ChaChaCha	Jive	Quickstep	Rumba	Samba	Tango	VienneseWaltz	Waltz
ChaChaCha	21	0	2	0	2	19	1	0
Jive	3	1	0	0	0	14	9	0
Quickstep	0	0	1	0	0	15	15	5
Rumba	5	1	0	4	0	4	19	7
Samba	11	0	2	2	0	4	11	2
Tango	0	0	0	0	0	22	18	0
VienneseWaltz	0	0	0	0	0	4	14	10
Waltz	0	0	0	2	0	0	14	32

Figure 14. Confusion Matrix for tempo augmentation = 0.7



Label \ Prediction	ChaChaCha	Jive	Quickstep	Rumba	Samba	Tango	VienneseWaltz	Waltz
ChaChaCha	-21	0	-1	0	1	16	1	0
Jive	1	-11	-4	-1	-1	14	7	0
Quickstep	-1	0	-29	-3	0	15	13	5
Rumba	5	0	-5	-26	0	4	14	3
Samba	11	-2	-2	2	-30	2	11	2
Tango	-2	0	-1	-1	0	-5	15	0
VienneseWaltz	0	0	0	-4	0	4	-12	10
Waltz	0	0	-1	0	0	0	4	-3

Figure 15. Subtracted Confusion Matrix for tempo augmentation = 0.7

Table 4. Ballroom dance recommended tempo

Dance	Tempo (bpm)
Cha-Cha-Cha	120 - 128
Jive	168 - 184
Quickstep	200 - 208
Rumba	100 - 108
Samba	96 - 104
Tango	120 - 140
Viennese Waltz	174 - 180
Waltz	84 - 90

4.4.2 Class confusion matrix

Fig.14 shows a confusion matrix of model's predictions on data with slower tempo transformation applied (rate=0.7), while the rest of the features remain the same. Fig.15 shows a difference between the original confusion matrix trained on non-transformed data, and a new confusion matrix produced on transformed data. If there is a positive number in the matrix off the diagonal, it means that a model has been classifying an example correctly before, but is now making mistakes. If there are negative numbers off the diagonal, it means the model became more accurate in that particular spot. Negative numbers of the diagonal mean mistakes and positive numbers mean improvements. For an irrelevant transformation, the best case scenario when we see a lot of 0s or near-0 numbers, which would mean that a model was not affected by the transformation. For a relevant transformation, we would want to check whether the effect is logical. For instance, among Viennese waltz, quickstep, rumba and samba, waltz is relatively slower. When decreasing tempo, we would assume other dances would be confused with waltz and see positive numbers in waltz's column. However, this is not what happens. Most dances get confused with Viennese Waltz or Tango. Tango is a medium tempo dance and Viennese Waltz is one of the faster dances, as shown in Table 4. Also, 10 Viennese Waltzes are now classified as Waltz and there are less Waltzes confused with Viennese Waltz then before, which is something that we would expect.

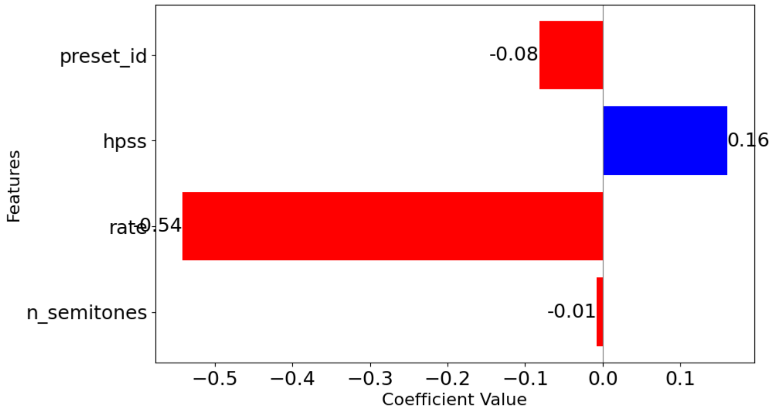


Figure 16. LIME used to explain a single prediction.

4.4.3 LIME explaining single instances

ATBE also implements the LIME technique for interpreting the MIR task model. The method centers on training Logistic Regression with Lasso regularization on transformed audio attributes, to estimate the forecasts of the CNN model underneath. Its objective is to comprehend the reasoning behind a specific prediction made by the CNN model, with the explanatory model reducing loss and complexity while closely imitating the original model's forecasts.

For instance, the Fig.16 demonstrates the coefficients the multi-class logistic regression model obtained for different audio transformation features, highlighting their effect on the model's prediction for a specific case. The $preset\_id$ corresponds to DRC, hpss to harmonic-percussive source separation, $rate$ signifies the tempo, and $n\_semitones$ indicates pitch. The most impactful feature is $rate$ (tempo) which is something we should expect.

In the context of music genre classification, implementing LIME for model interpretation can significantly enhance the understanding of how different audio features influence genre predictions. This approach is particularly valuable in academic research and development, where transparency in machine learning models is crucial.

5 Discussion and future work

Implementing interpretability methods like ATBE on a global scale in the field of MIR has the potential to revolutionize the way we understand and interact with deep learning models. By providing human-understandable explanations for model predictions, ATBE can enhance trustworthiness, transparency, and user experience in music-related applications. The ability to interpret the influence of various audio features on model predictions can lead to more accurate and reliable results, ultimately improving the overall performance of MIR systems. ATBE's approach of using automated augmented audio data to analyze model behavior under different conditions offers valuable insights into the decision-making process of deep learning models. By visualizing the impact of features like tempo, key, loudness, and harmonic/percussive source separation on model predictions, ATBE enables users to identify biases, detect errors, and understand the underlying mechanisms driving the model's decisions. This level of interpretability is crucial for ensuring fair treatment of users, improving model performance, and building trust in AI systems.

Moving forward, the implementation of ATBE on a global scale in the field of MIR could open up new avenues for research and development. Here are some potential directions for future work:

- **More transformations** Adding more transformations (e.g., remove certain instruments, modifying mode, chords, melody, style transfer). Developing efficient algorithms and tools for such transformations is possible, especially by using modern deep learning tools.

- **Integration with Existing Tools.** Integrating ATBE with existing interpretability tools and frameworks in the field of MIR could enhance the overall interpretability of deep learning models. By combining ATBE with techniques like SHapley Additive exPlanations(SHAP) and PDP, researchers can create more robust and comprehensive explanations for model predictions.

- **Ethical Considerations.** Addressing ethical considerations related to the use of ATBE in MIR applications, such as data privacy, bias detection, and algorithmic fairness, is essential for ensuring responsible AI deployment. By incorporating ethical guidelines and best practices into the development and implementation of ATBE, researchers can promote transparency and accountability in AI systems.

6 Conclusion

In conclusion, the field of MIR is essential for enabling access to music collections for listeners. Significant advancements in MIR tasks, such as music classification and recommendation systems, have been achieved through deep-learning models. However, the intricate nature of these models often presents challenges in understanding their decision-making processes, underscoring the importance of interpretability. The ATBE method, presented in this thesis, aims to improve the interpretability of deep-learning models in MIR tasks by offering explanations for model predictions using automated audio augmentations in AITAH, and employing PALUN to incorporate visualization techniques like confusion matrices and LIME plots. By enhancing transparency and comprehension of model decisions, ATBE can foster trust in AI systems within music applications and ensure equitable treatment of users. Ongoing research and advancements in enhancing interpretability within MIR models are crucial for progressing the field and enhancing user engagement in music-related applications.

Exploring the impact of different audio transformation on music classification accuracy is an promising tool for refining algorithms and systems that rely on robust MIR tasks. The ability to identify and address inaccuracies resulting from specific audio modifications ensures a seamless user experience and effective utilization of MIR technologies across various domains.

References

[AB12] Jean-Julien Aucouturier and Emmanuel Bigand. Mel cepstrum & ann ova: The difficult dialog between mir and music cognition. In *ISMIR*, pages 397–402, 2012.

[AJPBR^{+}24] Pablo Alonso-Jiménez, Leonardo Pepino, Roser Batlle-Roca, Pablo Zinemanas, Dmitry Bogdanov, Xavier Serra, and Martín Rocamora. Leveraging pre-trained autoencoders for interpretable prototype learning of music audio. *arXiv preprint arXiv:2402.09318*, 2024.

[Bah18] Hareesh Bahuleyan. Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*, 2018.

[BEH20] Adrien Bitton, Philippe Esling, and Tatsuya Harada. Neural granular sound synthesis. *arXiv preprint arXiv:2008.01393*, 2020.

[CFSK15] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim. Auralisation of deep convolutional neural networks: Listening to learned features. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, pages 26–30, 2015.

[CTMG23] Nikzad Chizari, Keywan Tajfar, and María N Moreno-García. Bias assessment approaches for addressing user-centered fairness in gnn-based recommender systems. *Information*, 14(2):131, 2023.

[DHP^{+}18] Rémi Delbouys, Romain Hennequin, Francesco Piccoli, Jimena Royo-Letelier, and Manuel Moussallam. Music mood detection based on audio and lyrics with deep neural net. *arXiv preprint arXiv:1809.07276*, 2018.

[dSMB21] Mila Soares de Oliveira de Souza, Pedro Nuno de Souza Moura, and Jean-Pierre Briot. Music tempo estimation via neural networks–a comparative analysis. *arXiv preprint arXiv:2107.09208*, 2021.

[DV17] Eero-Pekka Damskägg and Vesa Välimäki. Audio time stretching using fuzzy classification of spectral bins. *Applied Sciences*, 7(12):1293, 2017.

[EMO^{+}20] MFM Esa, NH Mustaffa, H Omar, NH M Radzi, and R Sallehuddin. Learning convolution neural network with shift pitching based data augmentation for vibration analysis. In *IOP Conference Series: Materials Science and Engineering*, volume 864, page 012086. IOP Publishing, 2020.

[FHP+22] Francesco Foscarin, Katharina Hoedt, Verena Praher, Arthur Flexer, and Gerhard Widmer. Concept-based techniques for" musicologist-friendly" explanations in a deep music classifier. *arXiv preprint arXiv:2208.12485*, 2022.

[FWVH23] Leonardo Fierro, Alec Wright, Vesa Välimäki, and Matti Hämäläinen. Extreme audio time stretching using neural synthesis. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[GKD+06] Fabien Gouyon, Anssi Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[HKVM20] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020.

[HMW20] Verena Haunschmid, Ethan Manilow, and Gerhard Widmer. audiolime: Listenable explanations using source separation. *arXiv preprint arXiv:2008.00582*, 2020.

[JHD23] David S Johnson, Olya Hakobyan, and Hanna Drimalla. Towards interpretability in audio and visual affective machine learning: A review. *arXiv preprint arXiv:2306.08933*, 2023.

[KMT08] Naoki Koshikawa, Takahiro Murakami, and Toshihisa Tanaka. Pitch shifting of music based on adaptive order estimation of linear predictor. In *Advances in Multimedia Information Processing-PCM 2008: 9th Pacific Rim Conference on Multimedia, Tainan, Taiwan, December 9-13, 2008. Proceedings 9*, pages 40–49. Springer, 2008.

[KN19] Sangeun Kum and Juhan Nam. Joint detection and classification of singing voice melody using convolutional recurrent neural networks. *Applied Sciences*, 9(7):1324, 2019.

[MHB15a] B. McFee, E.J. Humphrey, and J.P. Bello. A software framework for musical data augmentation. In *16th International Society for Music Information Retrieval Conference*, ISMIR, 2015.

[MHB15b] Brian McFee, Eric J Humphrey, and Juan Pablo Bello. A software framework for musical data augmentation. In *ISMIR*, volume 2015, pages 248–254. Citeseer, 2015.

[MKD18] Tobias May, Borys Kowalewski, and Torsten Dau. Signal-to-noise-ratio-aware dynamic range compression in hearing aids. *Trends in hearing*, 22:2331216518790903, 2018.

[MRL⁺15] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *SciPy*, pages 18–24, 2015.

[NNGY21] Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii. Audio-to-score singing transcription based on a crnn-hsmm hybrid model. *APSIPA Transactions on Signal and Information Processing*, 10:e7, 2021.

[PH17] Zdeněk Průša and Nicki Holighaus. Phase vocoder done right. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 976–980. IEEE, 2017.

[RD⁺22] K Sreenivasa Rao, Partha Pratim Das, et al. Melody extraction from polyphonic music by deep learning approaches: A review. *arXiv preprint arXiv:2202.01078*, 2022.

[RPM⁺23] Bruno Machado Rocha, Diogo Pessoa, Alda Marques, Paulo de Carvalho, and Rui Pedro Paiva. Automatic wheeze segmentation using harmonic-percussive source separation and empirical mode decomposition. *IEEE Journal of Biomedical and Health Informatics*, 27(4):1926–1934, 2023.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[SFR21] Mohamadreza Sheikh Fathollahi and Farbod Razzazi. Music similarity measurement and recommendation system using convolutional neural networks. *International Journal of Multimedia Information Retrieval*, 10:43–53, 2021.

[ŞJEC11] Umut Şimşekli, Antti Jylhä, Cumhur Erkut, and A Taylan Cemgil. Real-time recognition of percussive sounds by a model-based method. *EURASIP Journal on Advances in Signal Processing*, 2011:1–14, 2011.

[SP22] Arun Solanki and Sachin Pandey. Music instrument recognition using deep convolutional neural networks. *International Journal of Information Technology*, 14(3):1659–1668, 2022.

[SS18] Jérôme Sueur and Jérôme Sueur. What is sound? *Sound Analysis and Synthesis with R*, pages 7–36, 2018.

[SSB23] Mehrshad Saadatinia and Armin Salimi-Badr. An explainable deep learning-based method for schizophrenia diagnosis using generative data-augmentation. *arXiv preprint arXiv:2310.16867*, 2023.

[Stu14] Bob L Sturm. A simple method to determine if a music information retrieval system is a “horse”. *IEEE Transactions on Multimedia*, 16(6):1636–1644, 2014.

[TYS$^+$22] Ryo Terashima, Ryuichi Yamamoto, Eunwoo Song, Yuma Shirahata, Hyun-Wook Yoon, Jae-Min Kim, and Kentaro Tachibana. Cross-speaker emotion transfer for low-resource text-to-speech using non-parallel voice conversion with pitch-shift data augmentation. *arXiv preprint arXiv:2204.10020*, 2022.

[WCNY20] Yiming Wu, Tristan Carsault, Eita Nakamura, and Kazuyoshi Yoshii. Semi-supervised neural chord estimation based on a variational autoencoder with latent chord labels and features. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2956–2966, 2020.

[WFBS20] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models. *arXiv preprint arXiv:2006.00751*, 2020.

[Yük22] Mustafa Yüksel. Reliability and efficiency of pitch-shifting plug-ins in voice and hearing research. *Journal of Speech, Language, and Hearing Research*, 65(3):878–889, 2022.

[ZK21] Muhammad Rehman Zafar and Naimul Khan. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541, 2021.

Appendix

I. Configuration file example in AITAH

```yaml
# The path to save the augmented audio outputs
augmented_audio_save_path: audio/augments/wav

# The path to save the augmented metadata
augmented_meta_save_path: audio/augments/meta

# The path to save the data sources
mir_dataset_path: wavfiles_16k

# Harmonic/Percussive Source Separation
hpss:
  apply: true

# Logspace Time Stretch
tempo_factor:
  lower: -1
  upper: 1
  n_samples: 5

# Linear Pitch Shift
keys:
  n_samples: 5
  lower: -2
  upper: 2

# Dynamic Range Compression
drc:
  - "radio"
  - "film standard"
  - "film light"
  - "music standard"
  - "music light"
  - "speech"
```

II. Access to Source Code

ATBE, the Python package, can be found in this GitHub repository given as below: https://github.com/D3annyC/atbe

III. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Cheng-Han Chung**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Audio Transformations Based Explanations (ATBE) for deep learning models trained on musical data,

   supervised by Anna Aljanaki.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Cheng-Han Chung
14/05/2024